# AOZ Studio Beta - Bug #718

## Deek broken again.  (Fails entirely.)  Leek fails under certain conditions.

04/10/2021 10:06 AM - Brian Flanagan

| | | | |
|---|---|---|---|
| **Status:** | Feedback | **Start date:** | 04/10/2021 |
| **Priority:** | High | **Due date:** | |
| **Assignee:** | AOZ Developers | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0:12 hour |
| **Target version:** | | | |
| **Affected version:** | 1.0.0 (B8) | | |

**Description**

Deek was working.  Now it's failing again.  (Peek & Leek transpiling OK)
Deek is now transpiling incorrectly.

```
Peek(Buf+i) // transpiles to: aoz.peek(vars.Buf+vars.i)
Deek(Buf+i) // transpiles to: aoz.deek(%_ADDRESS)          <<== FAIL!
Leek(Buf+i) // transpiles to: aoz.leek(vars.Buf+vars.i)
```

### History

#### #1 - 04/10/2021 10:19 AM - Brian Flanagan

*- Assignee changed from Brian Flanagan to Francois Lionet*

*- Estimated time set to 1:00 h*

#### #2 - 04/10/2021 10:20 AM - Brian Flanagan

*- Estimated time changed from 1:00 h to 0:12 h*

#### #3 - 04/10/2021 12:20 PM - Brian Flanagan

*- Subject changed from Deek broken again. to Deek broken again.  (Fails entirely.)  Leek fails under certain conditions.*

This fails in both manifests.

NOTE: Make sure you delete the manifest.hjson in the resources folder if you want to easily test in both manifests.

Deek fails entirely due to incorrect transpilation.

Leek fails if the contents of a byte of the memory being examined goes beyond $7F.

The example below fails half-way through (when the high bit is set).

Code to test:

```
// #manifest: "amiga" ' Un-comment and delete manifest.hjson from resources folder to test Amiga manifest.

// Create & fill buffer
Reserve As Data $E,256
Buf = Start($E)
Siz = Length($E)
Print Buf,Siz
For i = 0 To Siz-1
    Poke Buf+i,i
Next i

// Peek Test
Print "Testing Peek():"
For i=0 To Siz-1
    Print Right$(Hex$(Peek(Buf+i),2),2);
    If (i+1) Mod 16 = 0 Then Print
Next i
Wait Key

/*
// Deek Test (Word Peek) - FAILS ENTIRELY
```

```
Print "Testing Deek():"
For i=0 To Siz-1 Step 2
    Print Right$(Hex$(Deek(Buf+i),4),4);
    If (i+2) Mod 16 = 0 Then Print
Next i
Wait Key
*/

// Leek Test (Long Peek) - Fails half-way thorugh.
Print "Testing Leek():"
For i=0 To Siz-1 Step 4
    Print Right$(Hex$(Leek(Buf+i),8),8);
    If (i+4) Mod 16 = 0 Then Print
Next i
Wait Key
```

In both manifests, Deek fails entirely.
In both manifests, Leek fails half-way through the test.

In the Amiga manifest, all 3 tests should be identical.
In the AOZ manifest, tests use big-endian.  Works with Leek until it fails.

#### #4 - 04/17/2021 12:54 AM - Brian Flanagan

*- Priority changed from Normal to High*

Re-tested in 1.0.0 (B7).  (rev. 4/16)
Errors still occur in both manifests.

**Deek(n)** still fails completely.
**Leek(n)** still fails for certain values.

I used the following to override the broken Deek command.  This works properly:

```
   //
  // Override default Deek that fails.
 // This override transpiles properly!
//
Function "Deek",_addr
 If Manifest$="amiga" // big endian
     result = Peek(_addr) << 8 | Peek(_addr+1)
 Else // little endian
     result = Peek(_addr+1) << 8 | Peek(_addr)
 End If
End Function( result )
```

#### #5 - 06/20/2021 07:35 PM - Brian Flanagan

*- Assignee changed from Francois Lionet to AOZ Developers*

*- Affected version changed from 1.0.0 (B7) to 1.0.0 (B8)*

Re-tested in 1.0.0 (B8) v14

Deek and Leek are still failing (in both manifests).

The fix in the notes for Deek works in both manifests.
I thought Leek had been fixed as well, but it's still not working.

#### #6 - 06/27/2021 02:21 PM - David Baldwin

The problem with the Leek part of your test program is that Hex$ doesn't like negative numbers, not Leek itself. Otherwise, this is fixed.

#### #7 - 06/27/2021 02:23 PM - David Baldwin

*- Status changed from New to Feedback*

#### #8 - 07/01/2021 02:01 PM - Brian Flanagan

*- Status changed from Feedback to Resolved*

re-tested in 1.0.0 (B8) u15
Deek is working now.
Leek is also working.  (Apparently, the bug here is in Hex$)

**#9 - 07/02/2021 01:10 AM - Brian Flanagan**

*- Status changed from Resolved to Feedback*

Hmmm... I take that back (about Leek). The problem is that these examples should not be negative numbers (at least not in AOZ mode). For the Amiga manifest, yes, but in AOZ mode, we should be able to use the longer integers.

The Hex$ function ***will*** handle the larger hex numbers just fine, BTW. (Up to $1FFFFFFFFFFFF)
NOTE: The range for large integers in AOZ is +/- 9007199254740991

There is also still a bug in the Hex$() function.